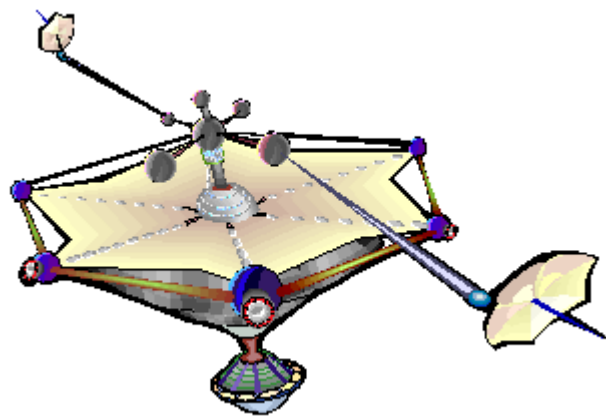


第五章 程式裡的 IO 控制

瞭解過程式的初始宣告與運算式等基本的元件以後，一般來說，當您要進行程式開發時，就必須瞭解操作者一定希望您所開發的程式能具有較強的人機對話能力（即操作親和力）。通常，高階語言都會提供了從輸入設備獲取輸入訊息以及用於螢幕與檔案輸出的函數。因此，您將在本章中學到：

- 輸出函數
- 輸入函數



5-1 輸出函數部分

本節將針對 Visual LISP 與 VBA 的輸出函數來為您做出一有系統的介紹。在練習前請注意以下的重要觀念事項：

1. 既然談到「輸出/輸入」，那就會牽涉到要將從哪裡將輸入讀進來，或將資料輸出到哪裡。最常見的情況就是：要從某一檔案讀入，或將資料寫入到某一檔案中。在這之中，檔案所在的路徑就很重要。雖然我們可以直接在程式中寫出：

```
"d:\\sample\\textfile.txt"
```

這樣的語法來表示要讀入的檔案完整路徑，但是這卻是不智的。因為萬一這個檔案未來改了路徑，那程式就要再改，這是很不好的寫法。在本電子書 2-6 節中，我們已經針對 AutoLISP/VLISP 部分，提供了一個解決方案來解決這個問題。因此，在 AutoLISP/VLISP 程式中應該盡量避

免含有磁碟代碼與目錄名稱的路徑指定。VBA 部分因尚無適當的方法，所以暫時只有指定完整路徑。

2. 讀入部分使用上述原則是沒錯的，但是輸出就可以指定完整路徑了。因為要將資料送到哪裡，的確會是一個變數。可是，就用 2-6 節的方法來做也未嘗不可。

5-1-1 用於螢幕和檔案輸出的 Visual LISP 函數

在前面章節的範例中，我們經常用到一個螢幕輸出的函數 — `princ`。`princ` 不但可以用來履行字串的螢幕輸出，也可以履行字串檔案的輸出。與 `princ` 功能類似的函數還有 `print` 和 `prin1`，它們的用法很類似。例如：

```
(setq f (open "textfile.txt" "w"))  
(print "Learning VLisp" f)  
(prin1 "Learning VLisp" f)  
(princ "Learning VLisp" )
```

其中，`f` 是檔案描述子，整個語法的意思就是：若有 `f`（即檔案若已開啓），則將字串寫入 `f` 所指定的檔案中；如果沒有開啓檔案，就將字串輸出列印於螢幕上。雖然 `princ`、`print` 與 `prin1` 的用法類似，但是這三個函數也有一些區別：

「`princ` 用於顯示不含引號的字串，並傳回字串；`prin1` 顯示的是包含引號的字串，並傳回字串；而 `print` 也是顯示包含引號的字串，並傳回字串，但是會在字串之前加一個空行，之後加一個空格。」

此外，`write-char` 和 `write-line` 也是兩個可用於螢幕輸出與檔案件輸出的 Visual LISP 函數。`write-char` 的作用是將一個字元寫到螢幕上或一個已打開的檔案中；而 `write-line` 則是針對字串來操作。它可以將字串寫到螢幕上或一個已打開的檔案中。舉例來說：

```
(write-char 65 f) ;到 f 指定的檔案中寫入字元 A，A 的 ASCII 碼是 65  
(write-line "d:\\sample") ;在螢幕上列印字串"d:\\sample"
```

5-1-2 僅用於螢幕輸出的 Visual LISP 函數

在 Visual LISP 函數中，prompt 與 terpri 函數是專門用於螢幕輸出的。prompt 的功能與 princ 類似，但傳回 nil。Terpri 的主要作用是換行，並傳回 nil。其功能就相當於加了列印控制子 "\n" 的 princ 函數：(princ "\n")。例如：

```
prompt("Learning Visual Lisp");在螢幕上列印字串"Learning Visual Lisp"  
(terpri);換行，並將游標移至下一行開始處
```

5-1-3 用於螢幕輸出的 VBA 函數 — prompt

在 VBA 中，如果希望在螢幕上列印文字，就可以使用「prompt 方法」。在以後提及的物件中，也稱為「prompt 方法」。例如：

```
Sub ForPrompt()  
thisdrawing.utility.prompt "learn VBA"  
End Sub
```

就是要執行 ForPrompt 巨集來於目前的 AutoCAD 指令提示區中印出 "learn VBA"。

5-1-4 用於檔案輸出的 VBA 函數 — print、write 與 writeline

print 是 VBA 內建的列印函數。其語法如下：

```
print [#filenumber] [outputstring]
```

其中，[#filenumber] 就是已打開的檔案編號，若包含此項參數，您就可以將 [outputstring] 中的內容順序寫入檔案中；若沒有包含 [#filenumber]，則將直接視窗中印出 [outputstring] 的內容。例如：

```
Open "d:/samples/textfile.txt" For Output As #1  
Print #1, "This is a test"  
Close #1
```

即是於 d:/samples/ 目錄下打開文件 textfile.txt 並寫入字串 "This is a test"。如果您直接輸入：

```
Print "This is a test"
```

則將直接在視窗中印出 "This is a test"。

write 函數則是用於撰寫一個指定的字串到一個 TextStream 的檔案中。所謂「TextStream」就是一個可順序存取的檔案流，這在後面的第十章中還會詳細講述。

writeline 函數則是按行寫入，即寫入一個指定的字串與換行到一個 TextStream 檔案中。例如：

```
Sub For fsoSample()  
Dim fso As new FileSystemObject  
Set fa=fso.CreateTextFile("d:\sample\textfile.txt")  
fa.WriteLine("For WriteLine test.")  
fa.Close  
End Sub
```

這段程式的意思就是：在 d:\sample\ 目錄下建立一個名稱爲 textfile.txt 的文字檔案，然後在其內寫入字串 "For WriteLine test"。在這個程式裡，我們使用了進階的檔案操作，如果沒有看懂這個程式或執行後不正確也沒關係。請先於此熟悉這個輸出函數，然後到第十章中再進行更進一步的瞭解。

5-1-5 MsgBox 方法

在 Windows 的程式設計中，我們還需經常使用訊息框來顯示一些訊息。VBA 提供了一個符合 Windows 風格，可用於訊息框輸出的 MsgBox 函數。它專門用來產生訊息框。例如，當您輸入以下的程式時：

```
MsgBox "A Convenient Way to Show Message"
```

執行後，就會出現一包含了 A Convenient Way to Show Message 的訊息框：



圖 5-1 經由 MsgBox 方法所發出的訊息框

有關 MsgBox 更詳細的使用資訊，請直接參閱 VBA 的線上說明文件。

5-2 輸入函數部分

本節將針對 Visual LISP 與 VBA 的輸入函數來為您做出一有系統的介紹。

5-2-1 Visual LISP 的輸入函數 — Get 系列函數

在 Visual LISP 裡，Get 系列函數包括：GetInt、GetDist、GetPoint、GetCorner、GetAngle、GetOrient、GetString 與 GetKword 等。它們負責讓使用者輸入數值、點、角度與字串等資料。Get 系列函數在執行中都會暫停以等待使用者輸入指定類型的資料，並傳回輸入的值。表 5-1 將中列出 Get 系列函數以及其所需的使用者輸入資料類型：

表 5-1 Get 系列函數的使用者輸入類型

函數名稱	使用者輸入類型
GetInt	等待使用者輸入一個整數值
GetReal	等待使用者輸入一個實數值
GetDist	等待使用者輸入或以螢幕上選擇的兩個點距離來傳回一個實數或整數值
GetPoint	等待使用者輸入或從螢幕上選擇一個點值
GetCorner	等待使用者輸入或從螢幕上選擇一個點值（視窗的對角點）
GetAngle	等待使用者輸入或由螢幕上選擇的點決定一個角度值，常用於傳回相對角度
GetOrient	等待使用者輸入或由螢幕上選擇的點決定一個角度值，常用於傳回絕對角度
GetString	等待使用者輸入一個字串
GetKword	等待使用者輸入一個由 initget 預定義的關鍵字

以下，我們就製作一個可以看出 Get 系列函數的使用方法。這個程式將讓使用者輸入實數的半徑、圓心以及顏色來畫一個圓：

```
(1);;;GetXXX function demo-----getxxx.lsp  
(2);;;function:Show Getxxx use and draw a circle  
(3)
```



```
(4)(defun c:getxxx(/ na nb nc pa pb aa ao)
(5) (setq na(getint "\nInput an integer:"))
(6) (setq nb(getreal "\nInput a real:"))
(7) (setq pa(getpoint "\nInput a point:"))
(8) (setq nc(getdist "\nInput a dist:"))
(9) (setq pb(getcorner pa "\nInput another point:"))
(10) (setq aa(getangle "\nInput an angle:"))
(11) (setq ao(getorient "\nInput another angle:"))
(12)(initget "Red Green Blue")
(13)(setq ck (getkword "\nEnter a color(Red/Green/Blue): "))
(14)(command "color" ck)
(15) (command "circle" pa nb)
(96))
```

執行方式：執行後，只要輸入 na、nb、pa、nc、pb、aa、ao、ck 值，並以 pa 點為圓心，nb 為半徑，將目前圖層的顏色設為 ck，並畫圓。

分析：除了 GetCorner 之外，Get 系列函數的語法都是類似的，均為(getxxx "message")。

第(9)行 GetCorner 是以一點為基點而拉出一橡皮拉線來獲得另一點。此語法一般用於選取區域時。請注意，在該語法中必須有一基點。

對第(10)行和第(11)行的 GetAngle 和 GetOrient 來說，也可在語法中設一基點，並在執行中等待使用者輸入另一個點。系統將根據這兩個點連線的夾角來傳回角度值。與 GetCorner 不同，此基點可有可無。

第 (12) 行使用了 initget 函數來設定關鍵字。請注意關鍵字是用大寫字母來表示的。第(13)行中將使用 GetKword 函數，於此，使用者將輸入 initget 所定義的關鍵字來表示顏色。

5-2-2 Visual LISP 的檔案與鍵盤輸入函數 — Read-Char 與 Read-Line

Read-Char 函數將從打開的檔案中或鍵盤的輸入中讀入一個字元，並將該字元轉換成十進位的 ASCII 碼值後傳回。例如：

```
(setq f (open "d:/samples/textfile.txt" "r"))
(setq ch (Read-Char f))
(setq ch2 (Read-Char))
```

這 3 行程式將於 d:/samples 目錄中打開檔案 textfile.txt，並讀取一個字元來賦

予 ch 變數，接著再從鍵盤緩衝區讀入一個字元。請注意：如果當鍵盤緩衝區內沒有東西時，(Read-Char) 函數就會暫停以等待使用者輸入。若輸入字元 "a"，則傳回其 ASCII 碼 97。

Read-Line 函數則用於從鍵盤或已打開的檔案中讀取一行字串。例如：

```
(setq f (open "d:/samples/textfile.txt" "r"))  
(setq str (Read-Char f))  
(setq str2 (Read-Line))
```

這 3 行程式將於 d:/samples 目錄中打開檔案 textfile.txt，並讀取第一行，然後將字串值賦予 str 變數，然後再從鍵盤緩衝區讀入一行字串並將其賦予 str2。

5-2-3 VBA 的輸入函數 — 使用 Util 來套用 VLISP 的 Get 系列函數

VBA 可以利用 Utility 物件來直接使用 Visual LISP 的 Get 系列函數。請參考以下範例：

```
(1)'GetXXX function demo-----getxxx.dvb  
(2)'function:Show Getxxx use and draw a circle  
(3)  
(4)Option Explicit  
(5)Public Sub getxxx()  
(6)  Dim na As Integer  
(7)  Dim nb As Double, nc As Double  
(8)  Dim aa As Double, ao As Double  
(9)  Dim pa As Variant, pb  
(10) Dim str As String  
(11) Dim ck As String, cmd1 As String, cmd2 As String  
(12) Dim circ As AcadCircle  
(13) Dim util As Object  
(14) Set util = ThisDrawing.Utility  
(15) na = util.GetInteger("Input an integer:")  
(16) nb = util.GetReal("Input a real:")  
(17) pa = util.GetPoint(, "Input a point:")  
(18) nc = util.GetDistance(, "Input a dist:")  
(19) pb = util.GetCorner(pa, "Input another point:")  
(20) aa = util.GetAngle(, "Input an angle:")
```

```

(21) ao = util.GetOrientation(, "Input another angle:")
(22) str = util.GetString(True, "Input a string:")
(23) util.InitializeUserInput 128, "Red Green Blue"
(24) ck = util.GetKeyword("Enter a color(Red/Green/Blue):")
(25) cmd1 = "(command ""color"")" & vbCr & ck & vbCr
(26) cmd2 = "circle" & vbCr & pa(0) & "," & pa(1) & vbCr & nb & vbCr
(27) ThisDrawing.SendCommand cmd1
(28) ThisDrawing.SendCommand cmd2
(29)End Sub

```

執行結果：本例輸出結果與 5-2-1 節的 Visual LISP 範例相同。

分析：程式的(15) ~ (24)行就是使用了 utility 物件來套用 Getxxx 函數的方法。如此，使用者就如同在 VLISP 裡一樣的輸入適當的值。其中，括弧中逗號之前沒有包含參數的，如 (, "Input another angle:")，就表示此參處要使用預設值。第(27)和(28)行將使用(25)和(26)行所設定的指令行字串來向 AutoCAD 發出指令。

5-2-4 VBA 的檔案與鍵盤輸入函數 — Read 與 ReadLine

Read 方法用於獲得從一個 TextStream 檔案中讀取指定數量的字元，並傳回所得到的字串。而 ReadLine 方法則用於獲得從一個 TextStream 檔案讀取一整行，並傳回得到的字串。舉例如下：

```

(1)'FileInput function demo-----fileinput(FSO).dvb
(2)'function:open a text file and get its content
(3)
(4)Option Explicit
(5)Public Sub FSOInput()
(6) Dim fso As New FileSystemObject, fil1 As file, ts As TextStream
(7) Dim str As String, ch As String, filename As String
(8) filename = "D:/(03M)Samples/Volume4/textfile.txt"
(9) Set fil1 = fso.GetFile(filename)
(10) Set ts = fil1.OpenAsTextStream(ForReading)
(11) str = ts.ReadLine
(12) ch = ts.Read(3)
(13) ts.Close
(14) Debug.print str

```


(15) Debug.print cn

(16)End Sub

執行結果：將文字檔案 "textfile.txt" 的第 1 行字串賦予 str，然後再將第 2 行的前三個字元賦予 ch。

分析：第(10)行將設定文字檔案流 ts。第(11)行將讀出 ts 所指定的文字檔案流的目前行，並將該字串賦予 str。此時，指標將移到第 2 行。第(12)行將讀出 ts 所指定的文字檔案流目前行(已到第 2 行)起的前 3 個字元賦予 ch。第(14)、(15)行程式在正式的程式中是不會寫的，但是本例是爲了證明 str 與 cn 變數均已擷取到應該取得的字元，所以就指定要將其內容顯示於即時運算視窗中。如圖 5-2 所示：

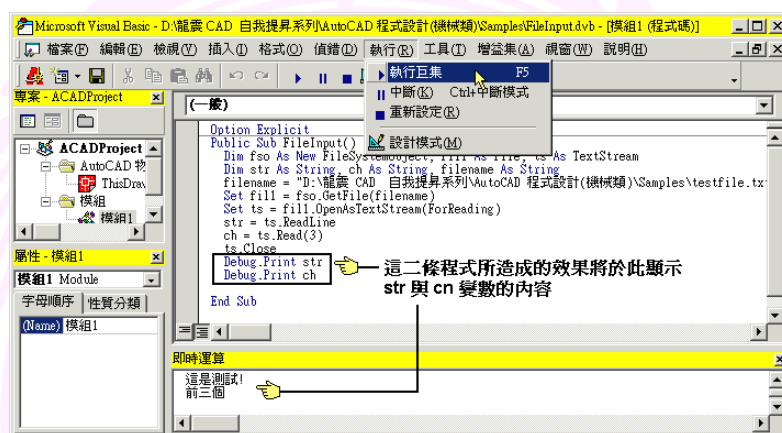


圖 5-2 str 與 cn 變數的內容

注意：本範例中有關 FSO 的語法部分，請參考 10-6 節。

5-2-5 InputBox 方法

與 MsgBox 輸出方法風格類似的輸入方法就是 InputBox。它將使用輸入視窗的方式來提示使用者進行輸入，非常直接好用。例如，輸入以下程式：

```
a = InputBox("Input number a, Please:")
```

就會出現以下的輸入視窗：

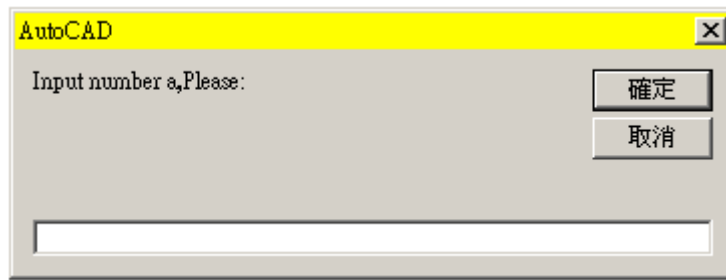


圖 5-3 經由 InputBox 方法所發出的輸入視窗

您只要在此輸入視窗中輸入參數即可。

啓發性習題

一.選擇題(單複選混合)

1.() 以下何者是用於螢幕輸出的 Visual LISP 與 VBA 函數？

- (a) prompt (VLISP) 與 print (VBA)
- (b) princ (VLISP) 與 prompt (VBA)
- (c) write-char (VLISP) 與 write (VBA)
- (d) write-line (VLISP) 與 writeline (VBA)

2.() 以下何者是用於檔案輸出的 Visual LISP 與 VBA 函數？

- (a) prompt (VLISP) 與 print (VBA)
- (b) princ (VLISP) 與 prompt (VBA)
- (c) write-char (VLISP) 與 write (VBA)
- (d) write-line (VLISP) 與 writeline (VBA)

3.() 以下何者是 MsgBox 的功用？

- (a) 專門用來建立訊息框
- (b) 專門用來建立輸入框
- (c) 專門用來建立下拉式選項
- (d) 以上皆非

4.() 以下何者是用於螢幕輸入的 Visual LISP 與 VBA 函數？

- (a) Read-char (VLISP) 與 read (VBA)
- (b) getxxx (VLISP) 與 Util + getxxx (VBA)
- (c) getxxx (VLISP) 與 getxxx (VBA)
- (d) Read-line (VLISP) 與 readline (VBA)

5.() 以下何者是用於檔案輸入的 Visual LISP 與 VBA 函數？

- (a) Read-char (VLISP) 與 read (VBA)
- (b) getxxx (VLISP) 與 Util + getxxx (VBA)
- (c) getxxx (VLISP) 與 getxxx (VBA)
- (d) Read-line (VLISP) 與 readline (VBA)

6.() 以下何者是 InputBox 的功用？

- (a) 專門用來建立訊息框
- (b) 專門用來建立輸入框
- (c) 專門用來建立下拉式選項
- (d) 以上皆非

二.實作問答題

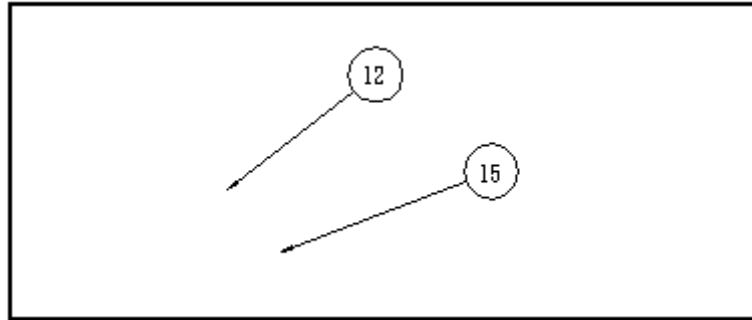
1. 請試設計一在繪製組立圖時，能讓我們輕鬆繪出指標圓球的程式（以 AutoLISP/VLISP/VBA 撰寫）。

設計詢問句：

1. 請指定箭頭尺寸: 20 <Enter>
2. 請指定指標圓球半徑: 50 <Enter>
3. 請指定文字高度: 30 <Enter>
4. 請點取欲放置箭頭的位置:
5. 請放置指標圓球位置:
6. 請指定指標圓球內的編號: 12 <Enter>

解答檔案名稱：BUBBLE.LSP，V-BUBBLE.LSP，BUBBLE.DVB

完成圖例：



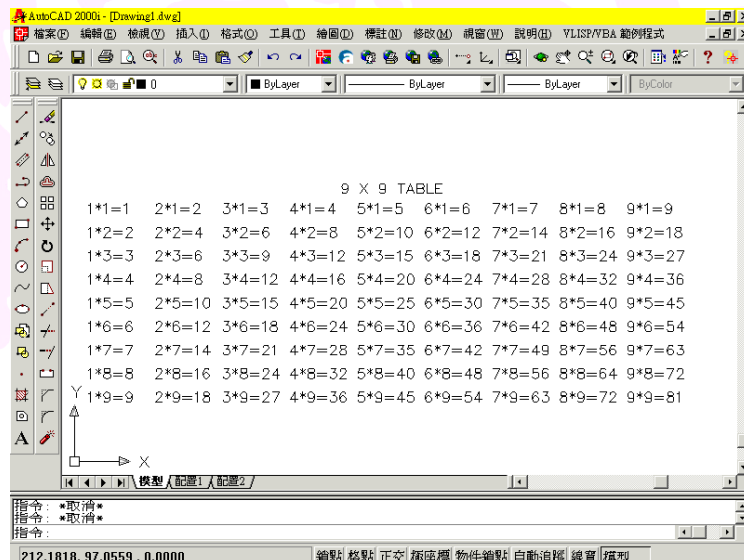
2. 請配合第七章的 While 迴圈語法，設計將寫於文字檔裡的九九乘法表到圖面上（以 AutoLISP 撰寫）。

設計詢問句：1. 請輸入文字起始點:

解答檔案名稱：99.LSP

配合檔案名稱：TAB99.txt

完成圖例：



3. 請配合第七章的 While 迴圈語法，設計一以輸入相對角度值來畫線的程式（以 AutoLISP 撰寫）。

設計詢問句：

起點：

輸入長度：50

輸入與前一點相對的角度：45

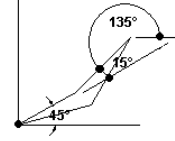
輸入長度 按 <C> 鍵表示欲頭尾相連,按 <Enter> 表示結束：50

輸入與前一點相對的角度：15

輸入長度 按 <C> 鍵表示欲頭尾相連,按 <Enter> 表示結束：50

輸入與前一點相對的角度：135

輸入長度 按 <C> 鍵表示欲頭尾相連,按 <Enter> 表示結束：C



解答檔案名稱：RLINE.LSP

